A Secure SSM Architecture

Ghassan Chaddoud, Isabelle Chrisment, and Abdelkader Lahmadi

INRIA - LORIA
615, rue du Jardin Botanique - B.P. 101
54602 Vandoeuvre-Les-Nancy - FRANCE
{chaddoud,ichris,lahmadi}@loria.fr,
WWW home page: http://www.loria.fr

Abstract. The SSM model is appeared in order to overcome the problems of deployment of IP multicast. However, a real commercial deployment of SSM have to offer some security services. Our work proposes an architecture, called S-SSM, for securing the SSM model. S-SSM defines two mechanisms for access control and content protection. The first one is carried out through subscriber authentication and access permission. As for the second, it is realized through the management of a unique key, called the channel key, k_{ch}, shared among the sender and subscribers. We have implemented a prototype of S-SSM in order to prove the feasability and evaluate the performance of our design.

1 Introduction

The IP multicast model [7] is appeared as a way to optimize the communication used by multimedia applications (audio and video conferences, video diffusion) involving several participants. But today, we can see that commercial multicast deployment is not a reality yet. Moreover, this model presents some limits such as scalable routing problems, address allocation problems, and notably security issues. Indeed, any host can send data to any IP multicast address and any host can join any group.

In order to overcome these problems, some simplified approachs have been proposed. Express [16] has defined a point-to-multipoint diffusion, and presents a group as the the tuple (S, E), where S is the unique sender and E the multicast destination address. The tuple (S, E) is called *channel*. Simple Multicast [3] is similar to Express but uses a birectional tree like CBT [2]. These different approaches, known as Source Specific Multicast (SSM) has been standardized in IETF with the proposition of PIM-SSM [14], version modified of PIM-SM [8] which allows source-filtering. Subscription to channels is supported by the version 3 of IGMP [4], called IGMP specific-source [15]. The SSM scheme presents a solution to address allocation, routing problems and a partial access control.

But, an effective commercial deployments of SSM should involve some security services such as access control and content provider protection. Having in mind the idea of ensuring SSM security, we propose in this paper a new architecture for securing an environment of multimedia diffusion, called S-SSM. The

diffusion environment is basically composed of SSM (PIM-SSM/IGMPv3) routing, video server, and potential receivers.

Our S-SSM architecture is composed of two security mechanisms: the access control and content protection. The access control mechanism is an extension of a solution proposed in [5] which uses a signed token to control access to group communication. The aim of such solution is to authenticate members by their local routers and to protect membership demands against attacks.

As for the second one, it is achieved via sender authentication and data ciphering. This last one requires the management of a unique key, called the channel key, k_{ch}, shared among the sender and subscribers. This scheme is a variant of Baal [5, 6] which is a scalable solution for the management of dynamic group keys.

In this paper, we present S-SSM, a secure architecture for an environment of 1-to-n multimedia diffusion. In Section 2, we define our general approach to secure an SSM environment. Then, in Section 3, we describe, more in details, the S-SSM architecture. Section 4 shows the preliminary performance tests achieved with our implementation. Section 5 presents related works. Finaly, we conclude with Section 6.

2 SSM Model

In our approach, we specify how securing an environment of multimedia multicasting. This environment is composed of (Fig. 2) :

- SSM (PIM-SSM/IGMPv3)[14, 15] routing. The version 3 of IGMP offers support for "source filtering", i.e., the possibility for a system to report interest in receiving packets only from specific sources.
- Video server or another type of applications like Internet TV, video-conference.
- Potential subscribers who have paid for a service.

In SSM communication, a multicast *channel* is defined as a datagram delivery service [16]. A channel is identified by a tuple (S, E) where E is a *channel destination address* [1]. Only the source host S may send to (S, E).

A *subscriber* host requests reception of data sent to a channel by explicitly specifying both S and E in a request, via an IGMPv3-report. The source S sends to a channel by simply transmitting datagrams addressed to E. The network layer guarantees that all datagrams sent by S to destination E are delivered to all subscribers of channel (S, E). The principal advantage of a such multicast diffusion service is a partiel access control to the channel :

- A source can not send data to a channel owned by other source. Contrary to the classic IP multicast model, the two channels (S, E) and (S', E) which have the same destination address E, are unrelated, despite the common destination address. Thus is guarantee by PIM-SSM routing.

[1] L'IANA has allocated the adress suffix $232.*.*.*/8$, *i.e,* 2^{24} class D addresses, for experimental use by the singl-source multicast model

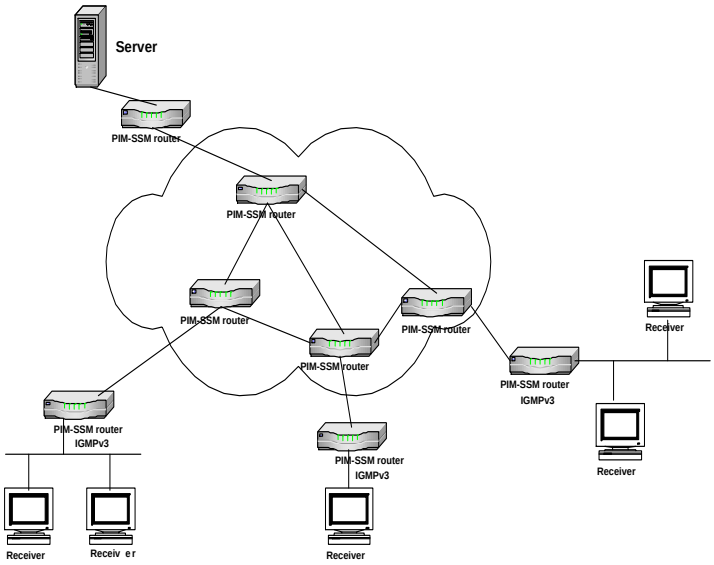

Fig. 1. Environnement SSM

– A subscriber to (S, E) will not receive data sent to (S', E) unless he subscribes to (S', E). Filtring by source guarantees that a subscriber does not receive all data sent with the same destination address E.

The SSM model appears appealing, but it has some limits when it is used by Internet access providers, especially, the contractors of multimedia diffusion services such as Internet TV, who require subscription fee for their services. An SSM model of security must allow only legitimate entities to access to channels for which they have payed subscription fee.

Having in mind the idea of the protection of SSM communication, we have defined S-SSM, a more complete security architecture, which offers two mechanisms of security : the access control and content protection :

– **Access control**. This mechanism aims at protecting network resources against malicious and illegitimate subscription requests. It is realized through :
  * Authentication of new subscribers when they subscribe to a channel.
  * Check of access permission to channel provided by the contractor when they subscribe to channel.
– **Content protection**. It is achieved via data ciphering. It requires the management of a unique key, called the channel key k_{ch}, shared among the sender and subscribers. Before diffusion, channel source encryptes data with the key k_{ch}. Only, subscribers having the key k_{ch} are capable of decrypting data. Thus, in addition to content secrecy, source authentication is ensured, because SSM routing forwards data coming only from the source.

We consider that the first mechanism would be an SSM extension by being integrated and inseperable part of subscription to channels. The second mechanism is independent from SSM and would be used, according to the security policy of service diffusion, by contractors who require secrecy for their content.

3 S-SSM architecture

In this Section, we describe further in details the S-SSM architecture and explain how these offered services are applied at the time of subscription and content diffusion.

3.1 S-SSM overview

Our **S-SSM** proposes that the actors of the diffusion environment, notably, the IGMPv3-capable routers, would be responsible for the security of the channel.

Two reasons explain this idea :

- The first one is to control some events that could compromise the network resources, in particular, the multicast routers. when one entity sends a subscription request, i.e, an IGMPv3-report, the first router receiving the request, must be capable to seal the request's fate. Indeed, if the entity meets certain conditions imposed by the channel security, the router can follow up the subscription request. Like that, we limit as much as possible network resource waste. Moreover, every malicious request is prevented from illegal access to channel flow.
- The second aspect is a scalability concern. The decentralized management of channel security is more flexible than the centralized one. In other words, instead of having only one entity controlling all the security operations within the channel, some entities distributed within domains where we have subscribers, can be delegated to do some security tasks. There are many advantages for this choice. The propagation of control messages will be limited to only domains where the subscription requests are issued. Moreover, the subscription latency to a channel is minimized and the bottleneck for the entity responsible for the security management is avoided.

So, the *channel manager* delegates the security tasks to the actors of the diffusion environnment. The set of actors cooperate through a group of communication, or a channel (GC, M) where GC is the address of the channel manager and M is an SSM address. This channel is called *control channel* or a signalization channel. This channel is responsible for the access control to diffusion channel and the management of the channel key k_{ch}.

The architecture S-SSM defines four security actors 2:

- **Global controller, GC**. It is an entity delegated by an authority responsible for data distribution such as content providers. According to information stored in a server, the global controller grants access to new subscribers and

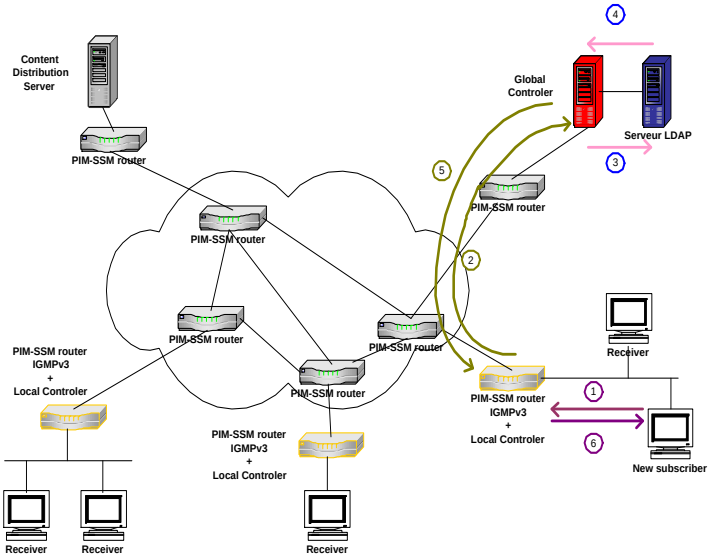

Fig. 2. S-SSM environment

revokes channel subscribers. It coordinates with *local controllers* in order to manage the channel key k_{ch}. In addition, the global controller tells the other controllers if there are any compromised subscribers in their domains. It rekeys the subscriber channel periodically. In addition, GC creates and manages the *control channel*.

- **Local controller, LC**. It is delegated by the global controller within its domain. It authenticates new subscribers and distributes k_{ch} to subscribers within its domains. It can, with the GC authorization, periodically rekey channel. A local controller can be implemented in an IGMPv3-capable router or an IGMP-proxy [9], which allows to extand delegation of the secured environment to a domain more large than a local area network. We call *channel controller* any local or global controller.
- **Directory Server, DS**. The service provider memorizes all information related to his clients or subscribers and needed to the management of the channel in a server, called *Directory Server*, DS. For each subscriber, DS contains a register or an entry per subscriber. An entry can be formed of several fields such as, *Validity* to prove that the entry is valid, *Start_date* and *End_date* beginning and ending date of the subscription, and specific nonce, Ns, attributed randomly to the subscriber. GC has access to DS only for reading. An example of a DS could be a LDAP server.
- **Subscriber**. Any entity which can receive channel flow and has payed, according to the service agreement with the service provider, subscription fee.

Prior to subscription, the new subscriber must have a valid entry in the DS server. The valid entry defines the *access permission.*

According to the server DS, the GC and local controllers create a global *recovery_list*, `RECOV_List`, composed of entities which had compromised or tryed to compromise the channel security. A member of this list has no more access to channel data. In addition, every LC has a list of local subscribers, `LOCALSUB_List`. This list is updated after every join or eviction of a subscriber. A LC distributes the key k_{ch} only to members of the list `LOCALSUB_SUB`.

In the remainder of this section, we show how to carry out the access control mechanism. We describe then the management of the channel key.

3.2 Access control

Before accepting subscribers to the channel, the global controller creates two keys for the control channel ; K_{ctl} key for controlling channel and K_{KEK} key for rekeying the first key. These keys are distributed to LCs. A local controller becomes delegated controller as soon as it has, after authentication by the global controller, these both keys.

In the following, we assume that a local controller is an SSM-capable router (IGMPv3/PIM-SSM) and that IGMPv3 defines a type of message which recognizes the definition of a signed *token.* The signed token forms an essential part of the authentication process of exchanged messages. A token is composed of:

* Ns specific number attributed by the service provider at the time of subscription and payement to the channel service,
* The tuple (S, ch), where S is the address of the source and ch is the channel destination address ;
* Timestamp,
* The IP address of the subscriber.

Remark For a signed token of a local controller, the feild Ns will be ignored or set to zero. It should be noted that the role of the token is to protect a non-secure subscription request against attacks.

As mentioned in Section 2, subscription phase is carried out in two steps:

- Authentication which is made by a local controller.
- Access permission check which is done by the global controller.

In the following paragraphs, we present the access control through the scenario depicted in figure 2, which shows the subscription of the host hh to a channel (S, ch). We assume that a local controller is on the SSM tree.

Authentication The entity, h, which wants to subscribe to the channel, sends a subscription message,$n°$ 1 in figure 2, composed of an IGMPv3-report with ch the address SSM of the channel and S the address of the channel source. This message is protected by the signed token of h. h sends the message to its

local controller. On receipt of this message, the LC authenticates the sender. We suppose that the LC knows the public keys of hosts within its domain. The subscription message is:

$h \rightarrow LC$: *IGMP-report*(S, ch), *[token_h]^ pK_h*

[token_h]^ pK_h is the h's token signed with its private key.

If the authentication succeed, the LC checks whether h is not in the `RECOV_List`. If not, it must verify with the GC whether h has right to access to channel (S, ch). If the authentication fails or if h is in the list `RECOV_List`, the message will be ignored. This step triggers the next step : validity check.

Validity check The purpose of this step is to verify with the `GC` whether the new subscriber has a valid entry in the `DS` server. The `LC` sends a message to `GC`, $n°$ 2 in figure 2. This message contains the tokens of `LC` and of the new subscriber. The message is:

$LC \rightarrow GC$: *[token_h]^ pk_h, [token_lc]^ pk_lc*

[token_lc]^ pK_lc is the lc's token signed with its private key.

By receiving this message, the GC authenticates first the `LC`. Then, if successful, it authenticates the host and checks with the `DS` server its Ns for the required channel and its period validity. This is done with the message $n°$ 3 of figure 2. This message is acknowledged by the `DS` with message $n°$ 4. If the answer is positive, the `GC` confirms the validity of the subscription by sending a message $n°$ 5 to the LC: $CG \rightarrow CL$: *[token_h]^ pK_h, [token_gc]^ pk_gc, Start_date, End_date*

This message contains the signed token of the host and the token of the GC and the both fields *Start_date, End_date* which indicates the beginning and the expiration validity of the subscription. In fact, these both fields forms with the subscriber's Ns the *access permission* to the channel.

At the reception of this message, the LC interprets this message as follow:

* If *End_date* $>$ *start_date* AND *TIME* $<$ *End_date* THEN the host can have access to channel data. The LC must start the phase of the distribution of the key of the k_{ch}.
* If *End_date* $>$ *Start_date* AND *TIME* $>$ *End_date* THEN the subscription demand is ignored.

3.3 Channel key management

In the case of channel diffusion, 1-to-n multicast, where there is only one sender, the channel key is used to ensure confidentiality and sender authentication. The channel rekeying should take place after:

– Reception of a new subscription request in order to avoid the subscriber from access to old channel traffic ;

- Access permission expiration or eviction of a subscriber in order to avoid him from access to futur channel flow;
- The periodicity of channel rekeying in order to avoid key forging.

In all cases, channel rekey is done by the GC. It creates a new key, k'_{ch}, and forms a message of rekey, *msg_rekey*. The message is composed of the channel address (S, ch), th GC's identity, the new key k'_{ch}, and two fields, *type* and *INFO*. The feild *type* is used to indicate the type of rekey. In the case of expiration of validity or eviction of subscribers, the field *INFO* contains informations about the evicted subscriber. The message *msg_key* is encrypted with the key of channel control, K_{ctl}, then diffused through the control channel to all LCs. When a LC receives a rekeying message, it decrypts this message and extracts the key then, in function to the field *type*:

* *Message of periodic rekey*, it distributes the new key k_{ch}' to all members of the list `LOCALSUB_List` with the old key k_{ch}.
* *Message after subscription*, if the new subscriber is not in their domain, it behaves as if it was a periodic rekey message. If not, it sends the new key encrypted the public key of the receiver, to all the members of the list `LOCALSUB_List` including the new subscriber.
* *Message after expiration of validity or explusion*, if the subscriber whose access permissoin validity has expired, is within its domain, first, it removes this subscriber from the list `LOCALSUB_List`. In the case of an expulsion message, all LCs must add the evicted subscriber to the list `RECOV_List`. Then, it sends the new key encrypted with the public key of the receiver, to all the members of the list `LOCALSUB_List` except the evicted one. If the expulsed subscriber does not belong to its domain, it can distribute the new key encrypted with the old one to all `LOCALSUB_List` members.

Remark The channel rekeying is, in all cases, carried out by the `GC`, but however, it can completely or partially delegate this task to `LCs`.

4 Implementation

This section uses implementation and discussion to study the performance of `S-SSM`. In first time, we show how we have implemented a prototype of the protocol and the topology we used to deploy it. Then, we measure the impact of the protocol, in term of latency, on subscription and data diffusion.

4.1 TestBed

In order to test our architecture, we have developped a testbed 3. The testbed is composed of 10 PCs working under FreeBSD 4.4.

The PCs are configured to work either as hosts or as routers. The host part of IGMPv3 is installed on all the hosts. *pimd*[2], PIM-SSM's daemon, is deployed

[2] http://catarina.usc.edu/pim/

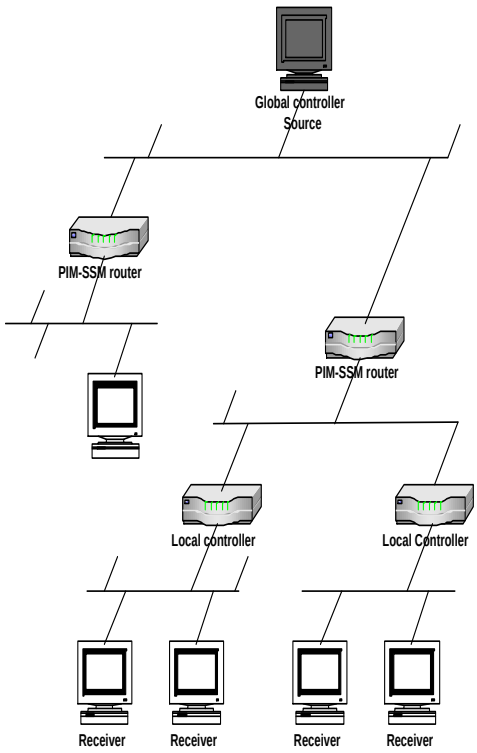

Fig. 3. Testbed

on the routers. We have modified *pimd* in order to recognize IGMPv3 reports and to update IGMPv3 table.

We have used two tools : TcpDump[3] to display the messages of IGMPv3 and our protocol. And Mtrace[4] to verify and check multicast messages of *pimd*. With these tools, we have run a simple multicast application to ensure the smooth running of PIM-SSM.

The cryptographic routines in our code are implemented using the publicly available OpenSSL[5]. OpenSSL offers many cryptographic functions such as RSA, DES, RC4, ... and key generation keys. Our protocol uses RSA [20] for authentication and key exchange encryption and DES [20] for key generation and content encryption. For more inforamtion about the performance of RSA and DES on machines used in the test, you can refer to [17].

[3] http://www.tcpdump.org
[4] http://www.http://www.free.net/ftp/packages/multicast/mtrace/
[5] http://www.openssl.org

4.2 Performance measures

subscription delay of new subscriber In the case of multimedia diffusion, protocol performances are measured in term of latency. So firstly, we have measured the subscription time of a new entity in three cases :

- non-secure subscription
- secure subscription with delegated controller and
- secure subscription without delegated controller.

Table 1 shows the delay between the sending of an IGMP-report and the receiving of the first packet of the channel. This table does not reflect the real delay because the testbed is limited to a LAN ; but it gives an representative idea about the facteur of delay at the time of the comparaison of the three cases.

	non-secure	delegated controoler	non delegated controller
subscription delay	0.088	3.361	6.822

Table 1. Subscriptoin delai of a new subscription (msec)

- We note that the latency, in the first case, is trivial by comparing to the two other cases. This result is not surprised because the subscription request sent under the form of a message IGMP-report, does not undergo any additional processing. Moreover, the local router forms part of the PIM-SSM tree. Consequently, the latency represents only the propagation time and the processing of the IGMP-report by the local router.
- As for the second case, we see that the time ot latency is more significant. This difference is explained by the implementation of the mechanism of access control with the new extension of subscription to a secured SSM tree. Indeed, the obtained delay represents the time of transmission and the processing of the message IGMP-report plus the necessary time for demanding the access permission with global controller. Consequently, the subscription latency depends stronly on the capacity of the link, the distance between local controller and the global controller, and the speed of processing the requests at the level of global controller.
- The latence in the third case is again more significant than the second case, because the local controller is not located on the secured diffusion tree, it must join the channel tree and the control channel.

To minimize the latency time induced by the secured subscription, we recommend :

* To increase the processing capacity of the global controller, *i.e,* powerful machine ;

* To delegate some tasks relating to channel key management to local controllers ; notably by allowing some local router to have access to the DS server ;
* To have many DS servers notably in domains where the number of subscribers is significant.

Rekey delay of channel key In the second test suite, we evaluate the delay of channel rekeying according to the channel size, *i.e,* number of subscribers. The delay of rekey is the time between the moment where the global controller decides to rekey the channel after subscription or expulsion and the moment where subscribers receive the new key. Figure 4 shows channel rekey for different channel size. We note from this graph that the time of distribution of a new key after a subscription or eviction of a member is constant and independant of group size because a controller needs only one multicast message to distribute the new key. Thus we show that our approach is scalable.

We can generalize the results of tests for channel of bigger size and, in consequence, deduce that in favour of the control channel, our approach make possible to solve the scalability issues for universal channel.

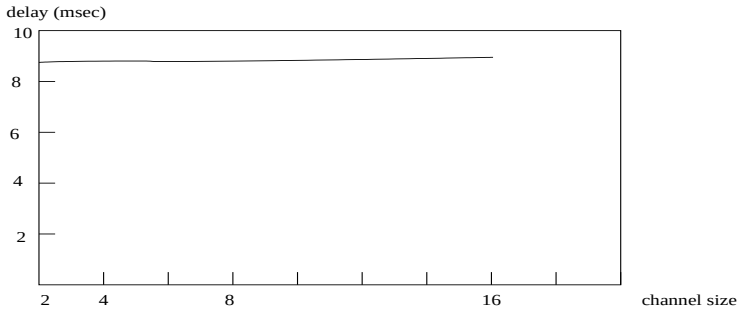

Fig. 4. channel rekeying delay

5 Related works

According to [13], securing IP multicast is divided into three parts :

1. Membership access control at the subnet level.
2. End-to-End data protection together with the group key management protocol.
3. Multicast routing protection (to ensure multicast control packets are authentic and thus preserves the correct routing behavior).

S-SSM presents a solution for membership access control and end-to-end content protection. If we consider that the protection of multicast routing is specific to each multicast routing protocol *i.e*, in our context, the responsibility of SSM routing protocols. Thus, our approach is a complete solution for securing SSM communication.

The use of signed token is not a new concept . First, we have used it in Baal [5, 6] to protect membership request in dynamic group communication and to authenticate new members, by local routers, before joining groups. [13] and [10] has used the token to provide access control through IGMP authentication. The access control is achieved via the provision of the token, as a proof-of-membership, to local multicast routers when hosts request joining to multicast trees.

In the context of dynamic group communication (DGC), many approaches key management, such GKMP [11, 12], LKH [22, 21, 19], or OFT [18, 1] can be integrated in our solution for the management of the channel key.

In the works done in Express [16], the authors have emphasized on interests of some aspects of security. They have proposed the use of the authentication for subscription requests : a host which wants to subscribe to a channel must provide, in addition to the adresses S and E, the key $K_{(S,E)}$. The subscription request propagates as long as the path toward the source, allowing the first router on the diffusion tree and receiving the request to authenticate the new subscriber because this router memorizes the key $K_{(S,E)}$. But this key is seen by routers as an optional parameter which allows to limit access to the channel. And the mechanism of key management is not explained here.

6 Conclusion

In this paper, we have presented a new architecture for securing SSM communication. The aim of this work is to allow only legitimate subscribers, i.e, subscribers who pay subscription fee, to have access to channel flow in a dynamic environment.

We have brought the security services on two phases. The first one is intended to ensure the subscription of a new host : we have used the access control, which is considered as a new extension to non-secure subscription to an SSM channel.

In the second phase, we have ensured confidentiality and sender authentication through the management of the channel key shared among the source and the subscribers of channels. We have conceived the architecture S-SSM of such a way that other approaches for the dynamic group key management such as GKMP, LKH or OFT could be used for the management of the channel key.

The preliminary results are encouraging et should be confirmed with local controllers delegated for domains more large with the implementation of IGMP-proxy [9].

References

1. D. Balenson, D. McGrew, and A Sherman. Key Management for Large Dynamic Groups: One-way Function Trees and Amortized Initialization, February 1999. Intenet draft: draft-balenson-groupkeymgmt-oft-00.txt.
2. A. Ballardie. Core Based Trees (CBT version 2) Multicast Routing, September 1997. RFC-2189.
3. T. Ballardie, R. Perlman, C. Lee, and J. Crowcroft. Simple scalable internet multicast, April 1999. tech. rep., University College London.
4. B. Cain, S. Deering, W. Kouvelas, and A. Thyagarajan. Internet group management protocole, version 3, January 2001. Intenet draft.
5. G. Chaddoud, I. Chrisment, and A. Schaff. Dynamic group communication security, May 2001. MMM-ACNS 2001, SaintPetersburg, Russia.
6. G. Chaddoud, I. Chrisment, and A. Schaff. Dynamic group key management, July 2001. ISCC2001, Hammamet, Tunisia.
7. S. Deering. Multicast routing in a datagram internetwork, Octobre 1991. Ph.D. thesis, Stanford University.
8. D. Estrin, D. Farinacci, and A. et al. Helmy. Protocol independent multicast sparse mode (pim-sm): Protocol specification, Juin 1997. RFC 2117.
9. B. Fenner, H. He, B. Haberman, and Sandick H. Igmp-based multicast forwarding (igmp proxying), November 2001. Intenet draft.
10. T. Hardjono and B. Cain. Key establishment for igmp authentication in ip multicast, octobre 2000. 1st IEEE European Conference on Universal Multiservice Networks (ECUMN'2000), Colmar France.
11. H. Harney and C. Mucknhirn. Group Key Management Protocol (GKMP) Architecture, July 1997. Request For Comments rfc-2094: Network Working Group.
12. H. Harney and C. Mucknhirn. Group Key Management Protocol (GKMP) Specification, July 1997. Request For Comments rfc-2093: Network Working Group.
13. H. He, T. Hardjono, and B. Cain. Simple multicast receiver access control, November 2001. Intenet draft.
14. H. Holbrook and B. Cain. Source-specific multicast for ip, May 2000. Intenet draft.
15. H Holbrook and B. Cain. Using IGMPv3 For Source-Specific Multicast, november 2001. draft-holbrook-idmr-igmpv3-ssm-02.txt, work in progress, IETF.
16. H. Holbrook and D. Cheriton. Ip multicast channels: Express support for large-scale single-source applications., November 1999. ACM SIGCOMM.
17. A. Lahmadi, G. Chaddoud, and I. Chrisment. Implementation d'un prototype du protocole baal. Rapport technique, LORIA, December 2001.
18. David A. McGrew and Alan T. SHerman. Key Establishment in Large Dynamic Groups using One-way Function Trees. TIS Labs at Network Associates, Inc. Gleenwood, Maryland, 1998.
19. S. Rafaeli, L. Mathy, and D. Hutchison. Lkh+2: An improvement on the lkh+ algorithm for removal operations, January 2002. Intenet draft.
20. B. Schneier. *Cryptographie Appliquée*. International Thomson Publishing, 1997. Traduction de L. Viennot.
21. D. Wallner, E. Harder, and R Agee. Key Management for Multicast: Issues and Architecture, September 1998. Intenet draft: draft-wallner-key-arch-01.txt.
22. C. Wong, M. Gouda, and S. Lam. Secure Group Communications using Key Graphs. ACM-SIGCOMM'98, septembre 1998.